Video-Based Interactive Media for Gently Giving Instructions

Takuya Kosaka†, Yuichi Nakamura‡, Yoshinari Kameda†, Yuichi Ohta†

† Department of Intelligent Interaction Technologies, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, 305-8573, JAPAN
‡ ACCMS, Kyoto University, Sakyo, Kyoto, 605-8501, JAPAN
(yuichi@media.kyoto-u.ac.jp)

Abstract. This paper introduces a novel multimedia system for instructing or guiding works. The system observes a user by image processing, and gives related information or appropriate advices by utilizing pre-recorded video archives. The distinctive feature of our media is that the system quietly observes a user and interrupts the user only when he/she really needs a help, for example, in a situation that the user asks a question. Otherwise, the system only presents related information that may be useful to the user, and it does not require any responses from the user. In this paper, a method for recognizing a user's status and a method for matching it to the contents in video archives are mainly described.

1 Introduction

Teaching a work has various aspects and various ways for it. An ideal way is an experienced human instructor: he/she tells how to do something, gives an advice, answers a question, just carefully watches what a student does or wants to do, or interferes if a student is about to make an irrevocable mistake. On the contrary, when we consider teaching or guiding a work by a conventional multimedia system, the system may ignore a student's situation. In other words, it may interfere and/or order the student to do exactly the same thing in stored data. We can think of, for example, a conventional system that teaches a way of cooking. The system will ask a user to do exactly the same things as shown in a recipe. Moreover, to activate QA function of the system, a user has to "ask a question" explicitly. In this sense, a QA system also forces a user to do something extra that interrupts his/her work. As seen in this example, it has not been well considered so far how a multimedia system should care the users and how it should not disturb them.

In this research, we propose a framework of video-based interactive media for giving instructions gently. Unlike conventional electronic instruction manuals, the system observes a user by image processing, and gives related information or appropriate advices by utilizing pre-recorded video archive. The distinctive feature of the system is that the system quietly observes a user and helps the user only when he/she really needs a help, for example, in a situation that the user asks a question. When the system recognizes that the user does not need any help, it only presents related information that may be useful to the user, and it does not require any responses from the user.

Currently, our target is a teaching system for assembly works on a desk top, and we are developing an experimental system for the task of assembling toy blocks. Although the system is still under development, we have implemented fundamental functions of the above framework: a method for matching a user's status to stored data as video manuals, a method for disambiguation, etc. We are currently going further toward fully automating this system and toward combining with question answering.

2 Video-Based Media not too Interfering

We propose our video-based interactive media that gently gives a user related information and appropriate instructions by observing users' actions. One of the most important functions is the recognition of a user's status. The other is the function of giving appropriate instructions only when they are necessary, since too frequent interferes or too much advices are annoying and even interrupt his/her works.

The fundamental technologies necessary for realizing the above framework are shown in Figure 1:

- Indexing to instruction videos
- Image recognition of objects and the user's actions
- Status matching between user's current status and indices of videos
- Presenting information relevant for the user's status

Among these, the method of video indexing that is done in advance is basically the same as that of QUEVICO[1], which delineates which information is required for which situation. For image recognition portion, we previously proposed our object tracking system for desktop works[2]. However, it needs to be improved for collecting sufficient user's information and object status, and we are currently developing the next system. This portion, therefore, is left for future work, and we currently gives manually collected results to our system.

On the other hand, we describe image recognition process of user's status in Section 3 in this paper. We implemented a status matching method, which continuously recognizes the user's current status by comparing the objects and actions of the user with indices of pre-recorded instruction videos.

The process is quietly performed inside the system, and it does not interfere the user as long as the system can recognize the status. In this case, related information is presented on a screen, and the system does not care whether the user watches it or not. On the contrary, when the user's status is out of a scenario, *i.e.* the video manual or it has too much ambiguity, the system inquires to the user for fixing the problem.

In the following sections, we will focus on the status matching and disambiguation.

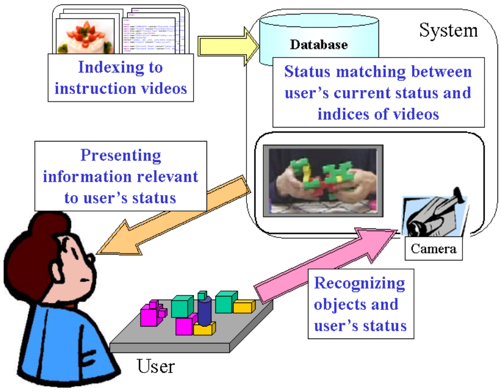

Indexing to instruction videos
Database
System
Status matching between user's current status and indices of videos
Presenting information relevant to user's status
Camera
Recognizing objects and user's status
User

Fig. 1. Video-based interactive media

3 Recognizing User's Status

3.1 Definitions and Description

For recognizing a user's status, continuous recognition of objects and the user's actions is essential[3], since direct estimation of the user's intentions is difficult. To simplify the recognition problem, we consider a work as a collection of tasks that are composed of primitive user actions and concerning objects. The followings define them:

Work: A work is a collection of tasks as shown in Figure 2, and it is also set as the goal of instruction.

Task: A task is a primitive function that is essential for assembly works. A task consists of action(s) and objects that appear in the task. We are currently using two types of task, "move an object" and "attach an object to another". Although other categories such as detaching or reshaping, will be surely necessary for future works, we are currently concentrating on these two because they are the most essential. The pattern of "attaching task" is shown in Figure 3, where, O_i means an object. A task is described by an ID, a name and concerning objects.

Action: An action is a primitive motion of a user, each of which is assumed to be recognized by image processing. We are currently using "lift an object", "place(put) an object" and "make two objects touched each other". An action is also described by an ID, a name, and concerning objects.

Object: An object is one of the components/parts that is visible and that has a concrete shape. It is described by an ID and the characteristics of visible features, *e.g.* color, shape, texture, etc.

Task graph: A task consists of a series of actions. Thus, the structure of a work is composed of a graph of tasks as shown in Figure 2.

Those data are given for each pre-recorded instruction video, and stored as indices. Since the cost of this indexing is not negligible, we expect that our object

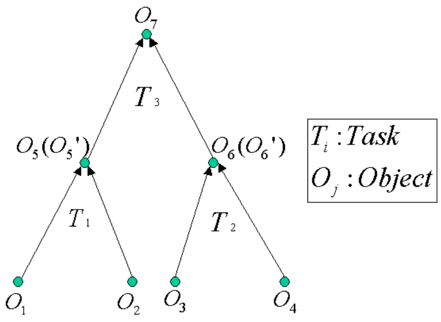


Fig. 2. Representation of a work

$attach\ O_1\ to\ O_2\ (and\ make\ O_3)$
$-lift\ O_1$
$-lift\ O_2$
$-make\ O_1\ and\ O_2\ touch\ (and\ make\ O_3)$
$-place\ O_3$

Fig. 3. Pattern of attaching task

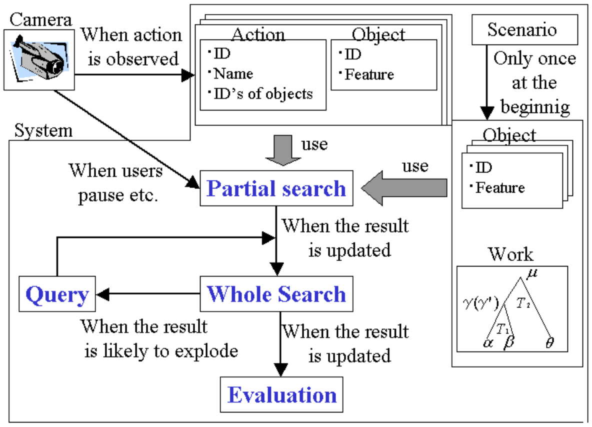


Fig. 4. Recognition of user's status

and action recognition system under development can be also available for this indexing process.

3.2 Status Recognition

Figure 4 shows the overview of the matching process.

(a) The indices of videos is input to the system, and the task graph, object data, are reconstructed from the indices of a video.
(b) When a user does something that can be recognized as an action, the corresponding action and concerning objects are recorded and added to the list of actions.
(c) The user's current status is recognized by comparing (a) and (b).

Step (c) is composed of "partial search" and "whole search" that will be described below.

Partial search: The system searches for a task, *i.e.* a set of consecutive actions, that matches a task in video indices. We use DP matching[4] between a sequence of user actions and a sequence of actions in video data, since a user does not always move exactly the same as recorded. All possible matches are searched, and the consistency among tasks are not considered in this step.

Whole search: The possible sequences of tasks are determined by checking consistency among objects and tasks. We use simple depth-first search for obtaining the possible combinations of tasks. Since the number of candidates suffers from combinatorial explosion, we need further mechanism for disambiguation by interacting with a user. This will be described in the next section. Each candidate for a task sequence is scored by using the similarity defined below.

For the matching step (c), the following criteria are used.

Similarity between objects $S(O_i, O_j)$: Similarity between objects is calculated based on object features such as color, shape, etc. Currently, we use the color histogram of an object region.

Similarity between actions $S(A_i, A_j)$: Similarity between actions is calculated by the product of "similarity between action names" and "similarity between concerning objects". Suppose an action $A_1(\mathrm{Name}_1, O_{11}, O_{12}, ..., O_{1n})$ and $A_2(\mathrm{Name}_2, O_{21}, O_{22}, ..., O_{2n})$, where $\mathrm{Name_i}$ means an action name and O_{ij} means a concerning object. The similarity between actions S(A_1,A_2) is calculated by the following formula.

$$S(A_1, A_2) = \delta(Name_1, Name_2) \cdot (\prod_{j=1}^{n} S(O_{1j}, O_{2j}))^{\frac{1}{n}} \tag{1}$$

$$\delta(X, Y) = \begin{cases} 1 \ (X = Y) \\ 0 \ (X \neq Y) \end{cases} \tag{2}$$

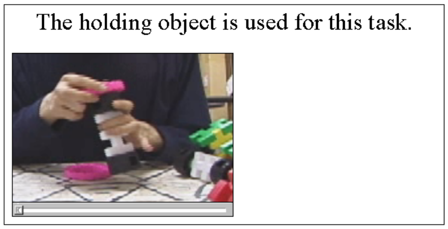
Similarity between tasks: Since it is difficult to recognize a task directly by image processing, the task the user performed is not directly matched to those in the video data. Tasks are recognized through comparing a sequence of actions in the above partial search process.

4 Interacting with Users

4.1 Data presentation to users

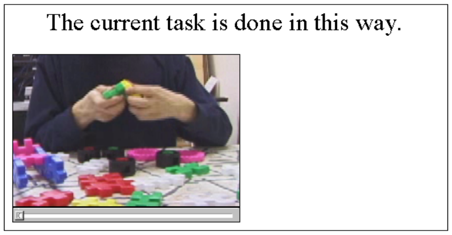
Video data relevant to an user's status, such as the explanation of the succeeding tasks, is presented without interfering his/her work. Basically, two kinds of data selection are considered.

Related to object: When a user keeps holding an object, or holds out an object toward the system, the system presents video data related to the object, *e.g.* a video segment that explains the operation that should be done to the object, a video segment that shows the usage with the object, and so on. An example is shown in Figure 5.



The holding object is used for this task.

Fig. 5. Presenting a video segment related to an object



The current task is done in this way.

Fig. 6. Presenting a video segment related to user's status

Related to user's status: When a user keeps moving and doing something, the system presents video data related to the current task. For example, the video segment that explains the current task or the next task required for the work. An example is shown in Figure 6.

In addition to the above cases, when a user asks a question to the system, the system will answer with appropriate video data. Although this QA mechanism is not implemented yet, it was previously proposed as QUEVICO[1] and it will be integrated in the near future.

4.2 Inquiry to users

Complete recognition of the user's status is a difficult problem even with the above mechanism. Interpretation of each action and an object is ambiguous, and the ambiguity of interpretation causes combinatorial explosion. We need additional mechanism for efficiently disambiguating the possible situations.

For this purpose, we are preparing a method of inquiring to the user. First, the system checks which portion is much ambiguous. The number of candidate objects or tasks in the videos is one of the indicator of the ambiguity. When the number of candidates are over a threshold value, the system inquires to the user about his/her status. By confirming an object name, a task name, or other things, considerable degree of ambiguity can be reduced. The followings are typical inquiry methods for objects and tasks.

Object: The system shows similar objects and asks the user to choose the correct one. For example, if one object held by a user has many candidates, *i.e.* objects in the video data, the system asks "what is the object in your hand?", or "which object is the same as the object you hold" by presenting objects list as shown in Figure 7. The user, for example, will choose the correct one by touch panel.

Task: The system shows a similar task and asks the user "are you doing this task?", "have you done this?", etc. An snapshot is shown in Figure 8**.**

The answer from the users are used for choosing the correct correspondence between a real object/task and a object/task in the instruction videos. Then status matching is performed again by using the correct correspondence of that portion.

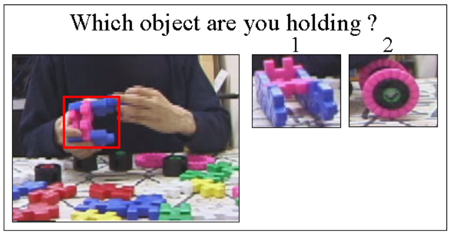

Which object are you holding ?

Fig. 7. Inquiring about an object

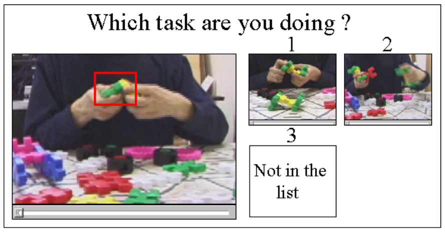

Which task are you doing ?

Fig. 8. Inquiring about a task

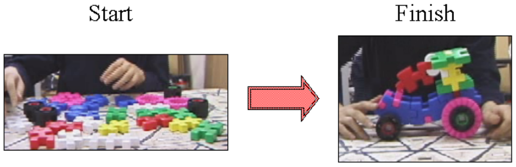


Fig. 9. Assembly of a toy block car

5 Experiments

For checking the potential of this system, we conducted preliminary experiments. An instruction video is taken for an actual assembly of a toy block car as shown in Figure 9. The toy block car is composed of 50 blocks and the work consists of 30 tasks. In another video, we also recorded the behaviors of a person who was asked to make the toy car by watching the video. From both videos, objects and motions are manually detected and given to the system. From the video of user's behaviors, about 70 actions are detected.

As a result of the first partial search, 90% of the tasks are correctly detected with false alarm of 95%. And figure 10 shows the precision and the recall rate of this experiment.

Figure 11 shows objects and the number of candidates for some objects detected during the experiment. Num(1) column shows the number of candidates without inquiries for disambiguation. The whole search for interpreting current status is not possible at this step, since each portion has too much ambiguity. Therefore, the system executed disambiguation by user inquiry. As object ID9 has 22 interpretations cadidates, it is the most ambiguous object. By inquiring to the user, the system obtained the correct matching between the objects. In this case, after obtaining the answer from the user, the improved result was shown at Num(2) in Figure 11. By this disambiguation, the interpretation candidates of object ID7 is also reduced, since the object ID7 and ID9 are used in the same task.

By repeating this type of inquiry, the system eventually gets a small number of candidates that include correct one.

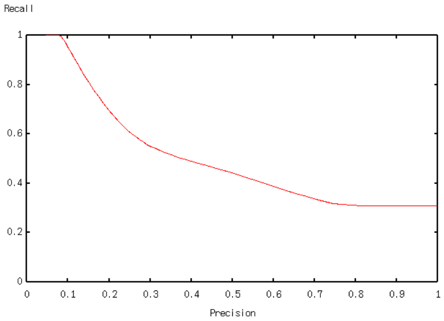
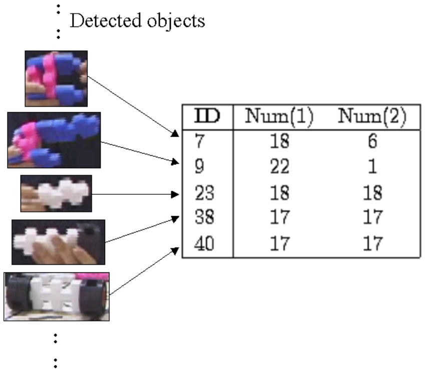


Fig. 10. Precision-recall graph


Detected objects

ID	Num(1)	Num(2)
7	18	6
9	22	1
23	18	18
38	17	17
40	17	17

Fig. 11. The number of the matched objects

6 Conclusion

In this paper, we introduced our idea of video-based interactive media that gently supports users. We proposed the framework for recognizing user's status and the method for reducing ambiguity by inquiring to users. In our preliminary experiments, this system succeeded in handling a toy-car assembly work in which 50 objects are used and 30 primitive tasks are required.

Our system is, however, still under development. To realize a realtime system, we need further intensive works. Integration with image processing portion is the most urgent topic. Building a good user interface is also an important topic.

References

1. Hidekatsu IZUNO,Yuichi NAKAMURA,Yuichi OHTA: QUEVICO:A Framework for Video-Based Interactive Media. Proc. Int'l Workshop on Intelligent Media Technology for Communicative Reality (2002) 6-11
2. Masatsugu ITOH,Motoyuki OZEKI,Yuichi NAKAMURA, Yuichi OHTA: Simple and Robust Tracking of Hands and Objects for Video-based Multimedia Production. IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems (2003) 252-257
3. Cen Rao,Mubark Shah: A View-Invariant Representation of Human Action. Control, Automation, Robotics and Vision (2000)
4. Seiichi Nakagawa: Pattern Information Processing. Maruzen (1999)